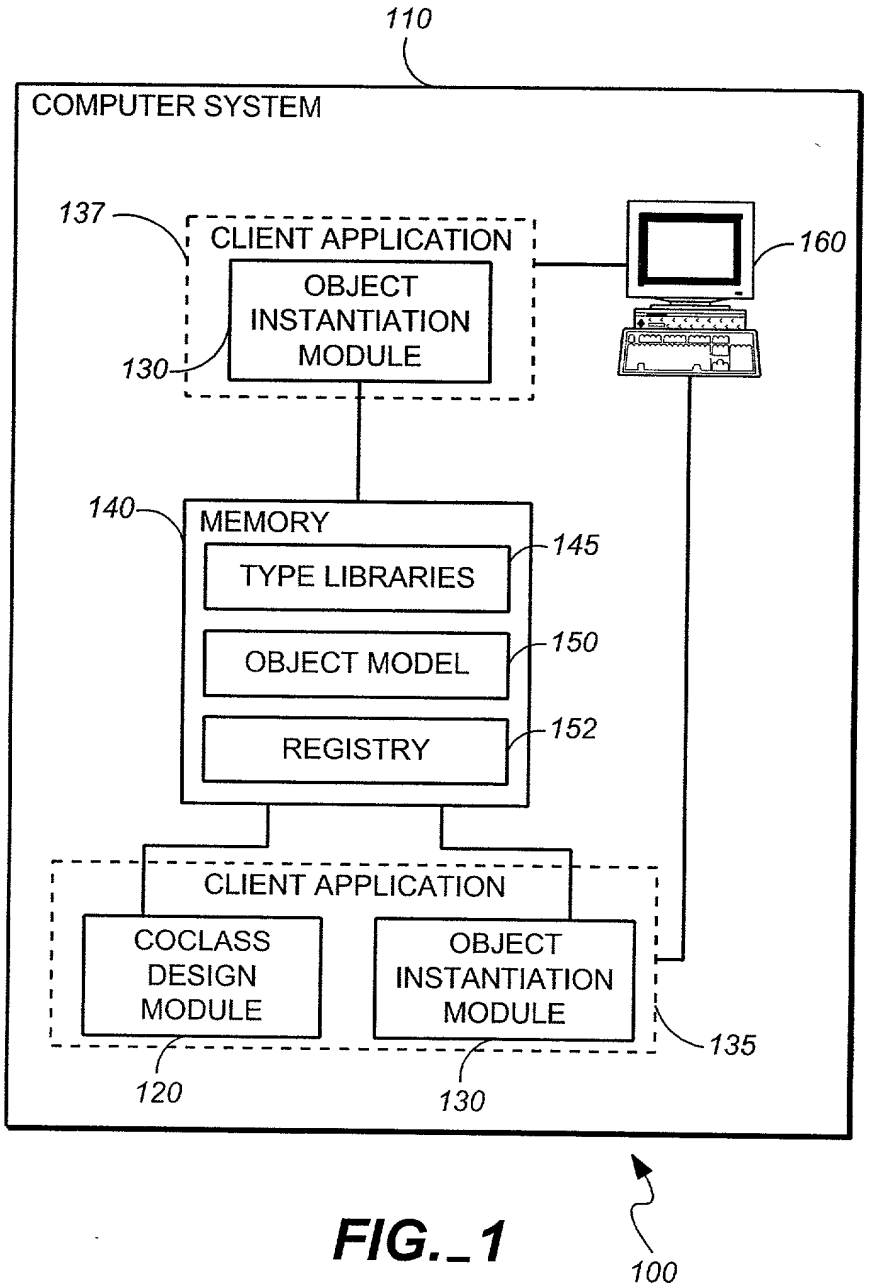

110
COMPUTER SYSTEM
137
CLIENT APPLICATION
OBJECT INSTANTIATION MODULE
130
160
140
MEMORY
145
TYPE LIBRARIES
150
OBJECT MODEL
152
REGISTRY
CLIENT APPLICATION
COCLASS DESIGN MODULE
OBJECT INSTANTIATION MODULE
135
120
130
100

FIG._1

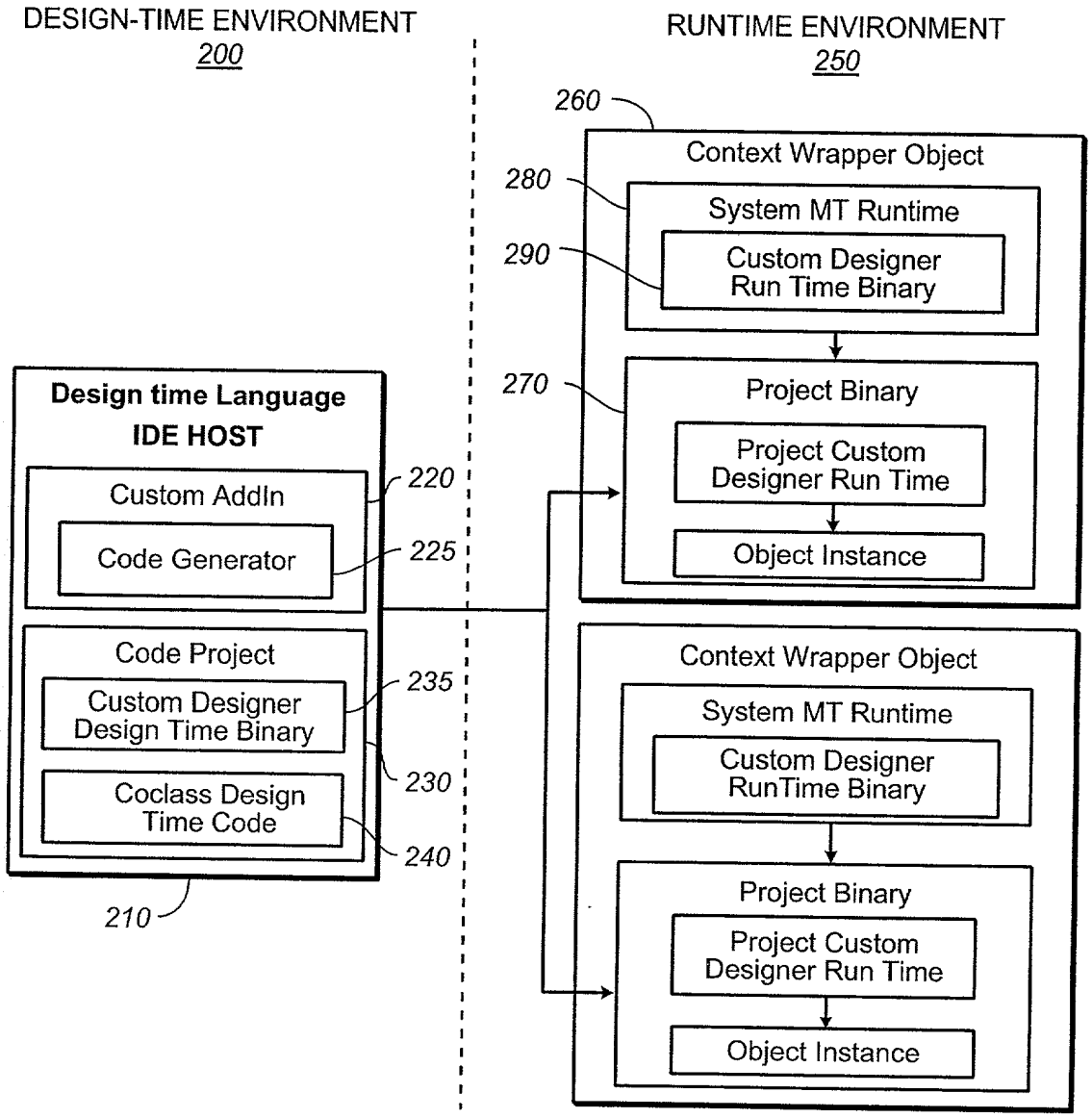

DESIGN-TIME ENVIRONMENT
200
RUNTIME ENVIRONMENT
250
260
Context Wrapper Object
280
System MT Runtime
290
Custom Designer
Run Time Binary
270
Project Binary
Project Custom
Designer Run Time
Object Instance
Design time Language
IDE HOST
Custom AddIn
220
Code Generator
225
Code Project
235
Custom Designer
Design Time Binary
230
Coclass Design
Time Code
240
210
Context Wrapper Object
System MT Runtime
Custom Designer
RunTime Binary
Project Binary
Project Custom
Designer Run Time
Object Instance

FIG._2

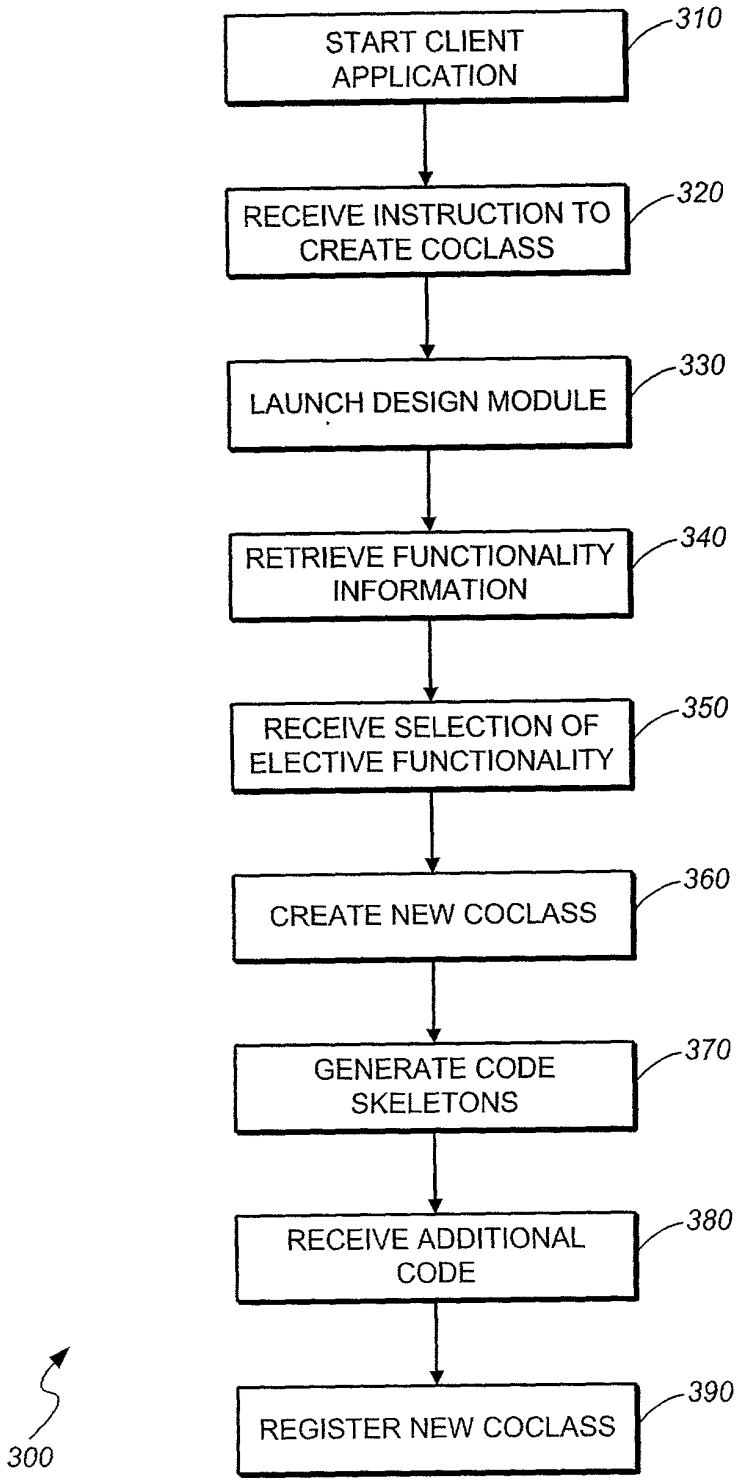

START CLIENT APPLICATION
310
RECEIVE INSTRUCTION TO CREATE COCLASS
320
LAUNCH DESIGN MODULE
330
RETRIEVE FUNCTIONALITY INFORMATION
340
RECEIVE SELECTION OF ELECTIVE FUNCTIONALITY
350
CREATE NEW COCLASS
360
GENERATE CODE SKELETONS
370
RECEIVE ADDITIONAL CODE
380
REGISTER NEW COCLASS
390
300

FIG._3

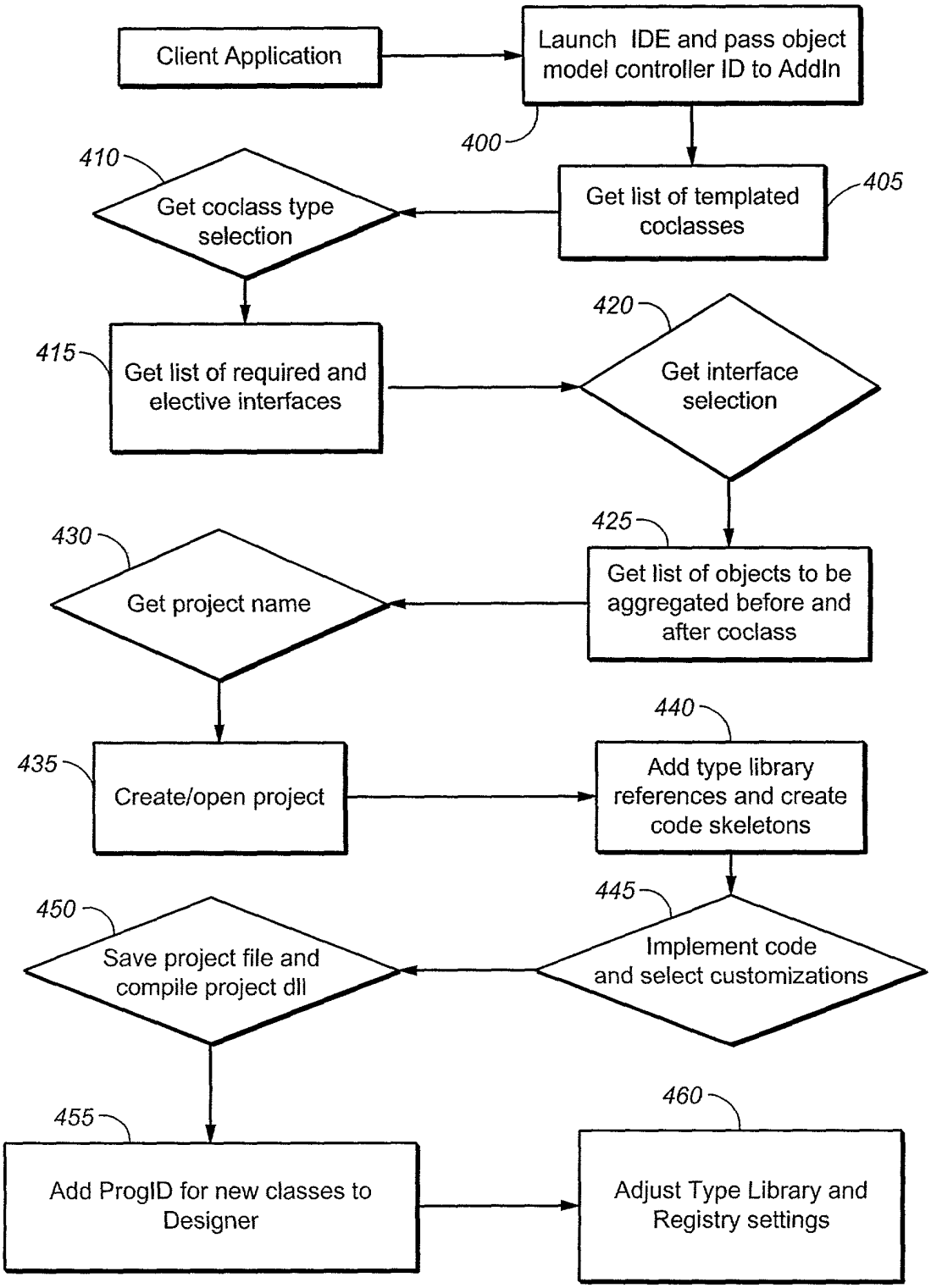

Client Application
Launch IDE and pass object model controller ID to AddIn
400
Get list of templated coclasses
405
410
Get coclass type selection
415
Get list of required and elective interfaces
420
Get interface selection
425
Get list of objects to be aggregated before and after coclass
430
Get project name
435
Create/open project
440
Add type library references and create code skeletons
445
Implement code and select customizations
450
Save project file and compile project dll
455
Add ProgID for new classes to Designer
460
Adjust Type Library and Registry settings

FIG._4A

"DEFT" 09872608

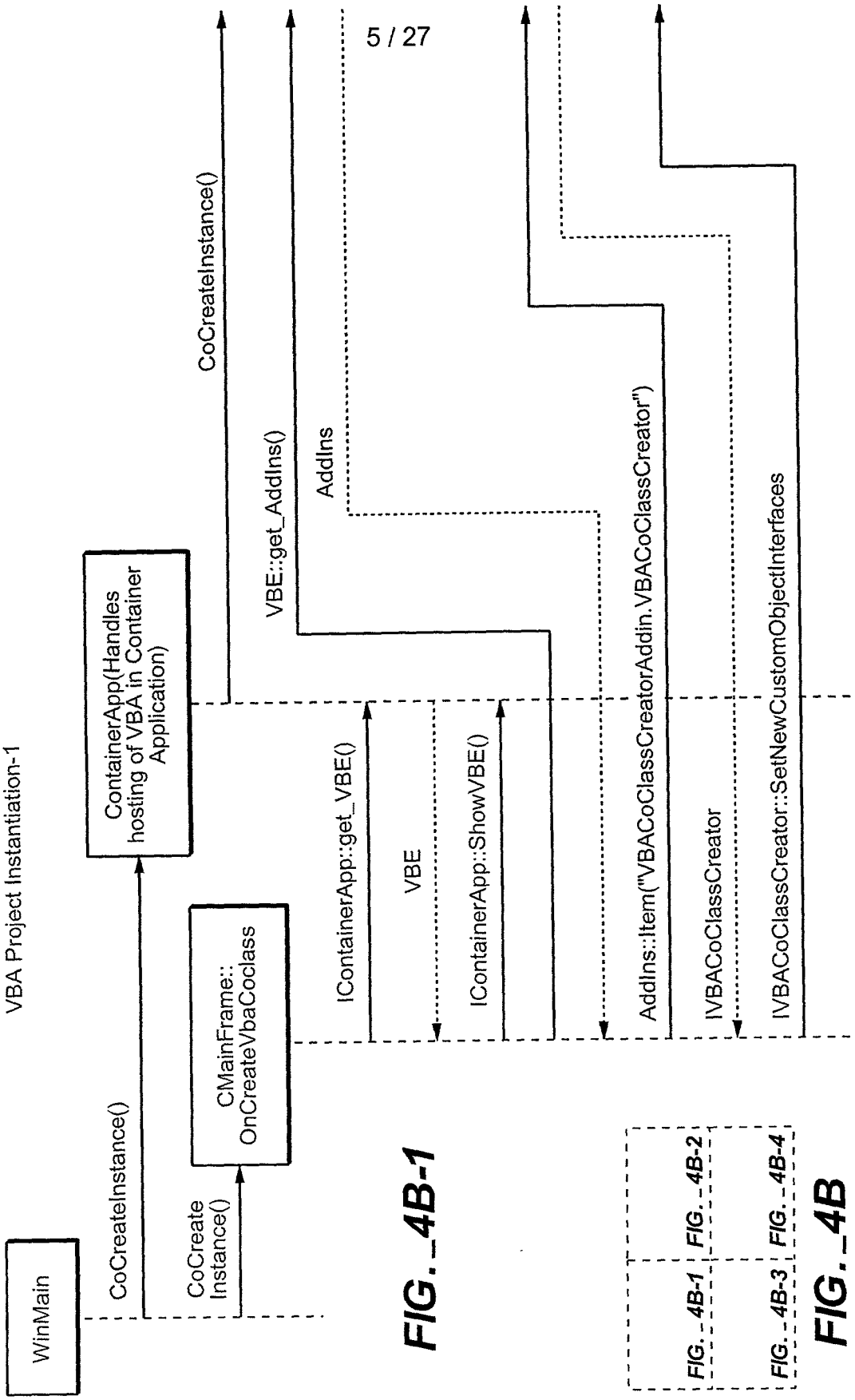


FIG._4B-1

FIG._4B-1 FIG._4B-2
FIG._4B-3 FIG._4B-4

FIG._4B

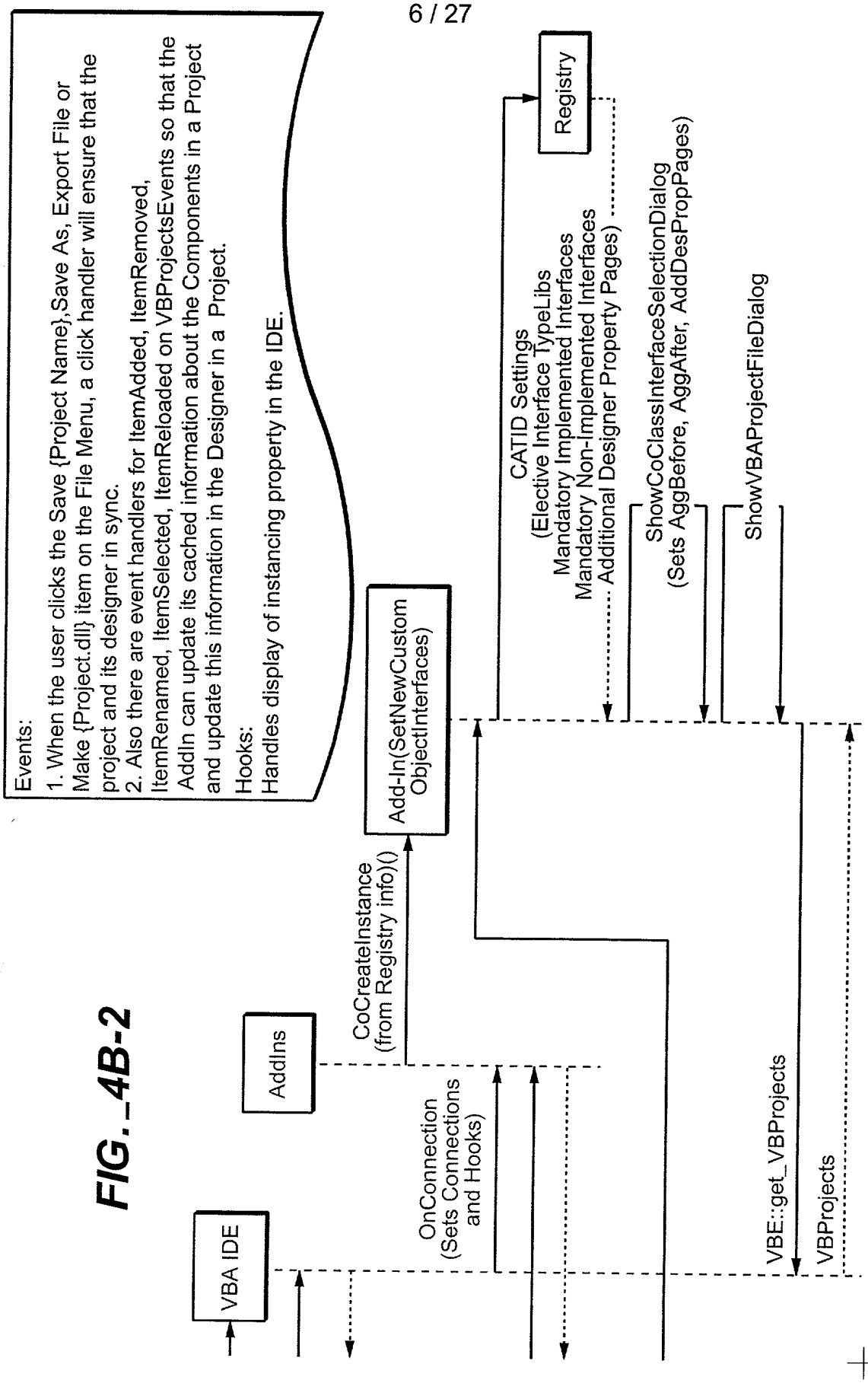

Events:
1. When the user clicks the Save {Project Name},Save As, Export File or Make {Project.dll} item on the File Menu, a click handler will ensure that the project and its designer in sync.
2. Also there are event handlers for ItemAdded, ItemRemoved, ItemRenamed, ItemSelected, ItemReloaded on VBProjectsEvents so that the AddIn can update its cached information about the Components in a Project and update this information in the Designer in a Project.
Hooks:
Handles display of instancing property in the IDE.
VBA IDE
AddIns
CoCreateInstance (from Registry info)()
Add-In(SetNewCustom ObjectInterfaces)
OnConnection (Sets Connections and Hooks)
CATID Settings (Elective Interface TypeLibs Mandatory Implemented Interfaces Mandatory Non-Implemented Interfaces Additional Designer Property Pages)
Registry
ShowCoClassInterfaceSelectionDialog (Sets AggBefore, AggAfter, AddDesPropPages)
ShowVBAProjectFileDialog
VBE::get_VBProjects
VBProjects

FIG._4B-2

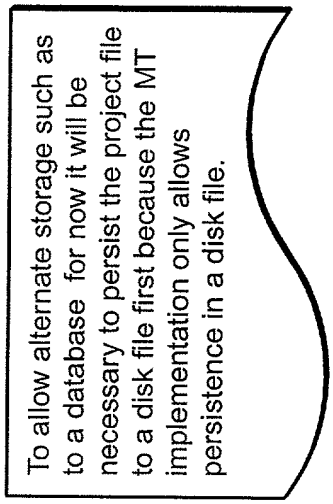

To allow alternate storage such as to a database for now it will be necessary to persist the project file to a disk file first because the MT implementation only allows persistence in a disk file.

FIG._4B-3

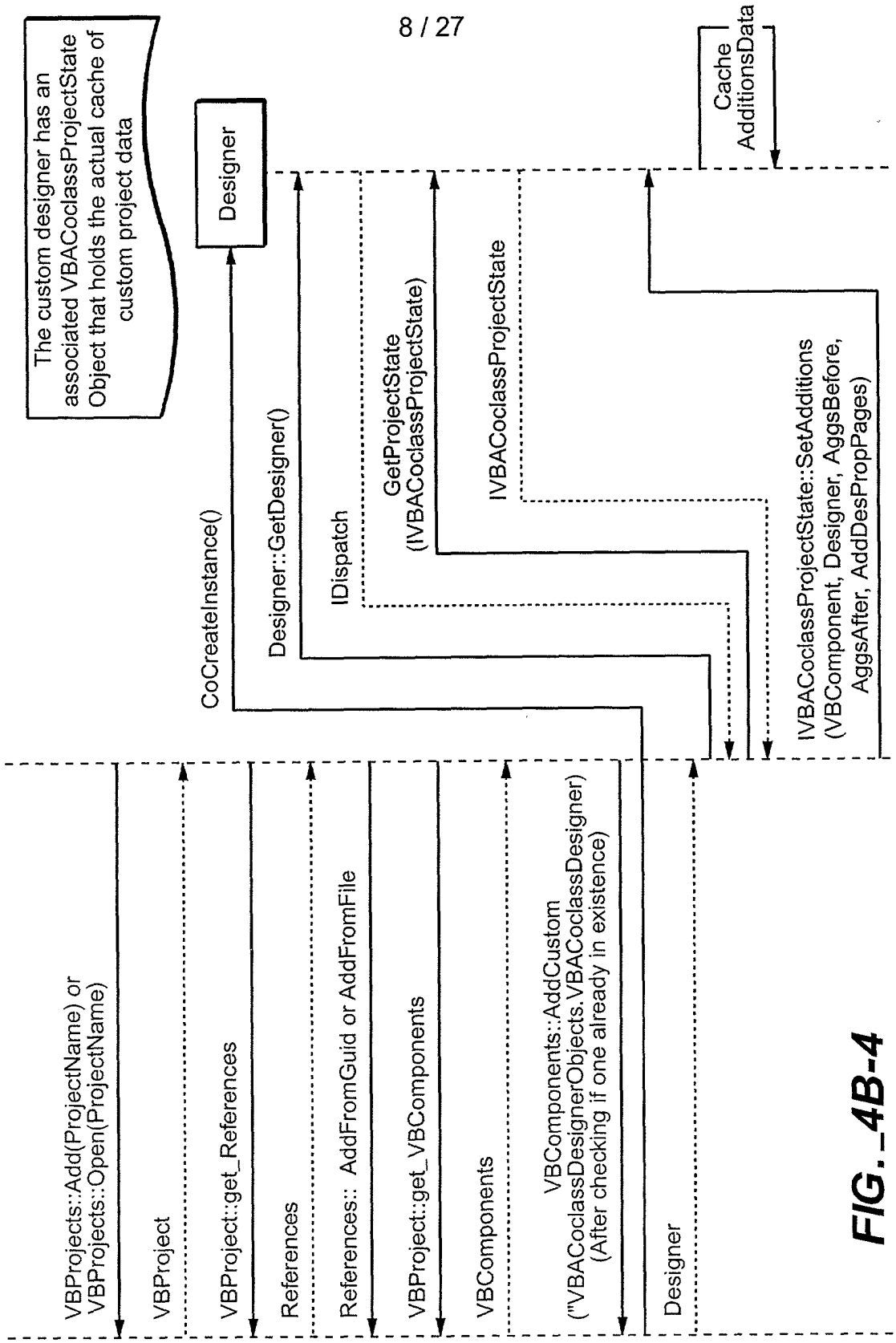

The custom designer has an associated VBACoclassProjectState Object that holds the actual cache of custom project data
Designer
Cache AdditionsData
CoCreateInstance()
Designer::GetDesigner()
IDispatch
GetProjectState (IVBACoclassProjectState)
IVBACoclassProjectState
IVBACoclassProjectState::SetAdditions (VBComponent, Designer, AggsBefore, AggsAfter, AddDesPropPages)
VBProjects::Add(ProjectName) or VBProjects::Open(ProjectName)
VBProject
VBProject::get_References
References
References:: AddFromGuid or AddFromFile
VBProject::get_VBComponents
VBComponents
VBComponents::AddCustom ("VBACoclassDesignerObjects.VBACoclassDesigner") (After checking if one already in existence)
Designer

FIG._4B-4

VBA Project Instantiation-2

9 / 27

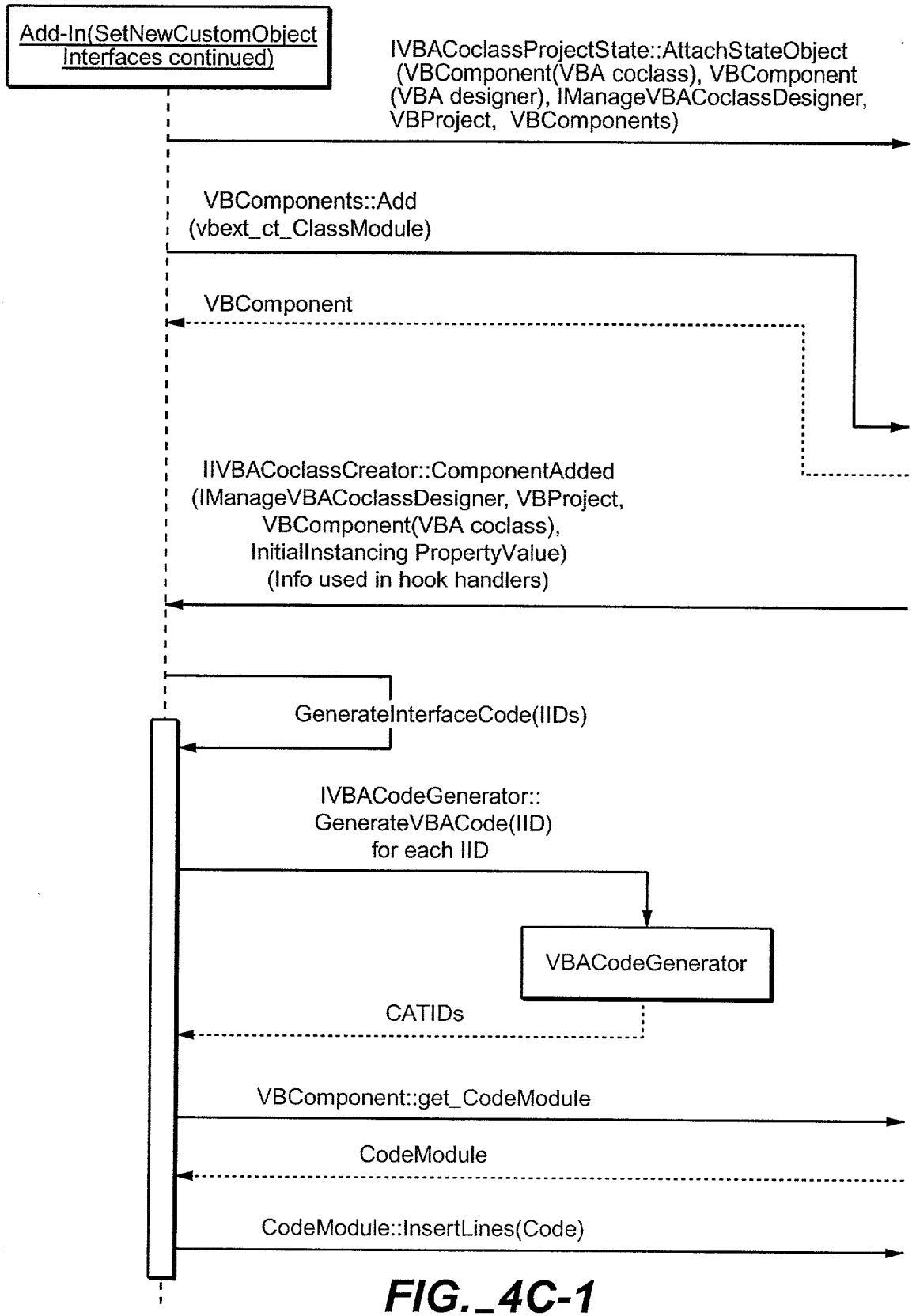


FIG._4C-1

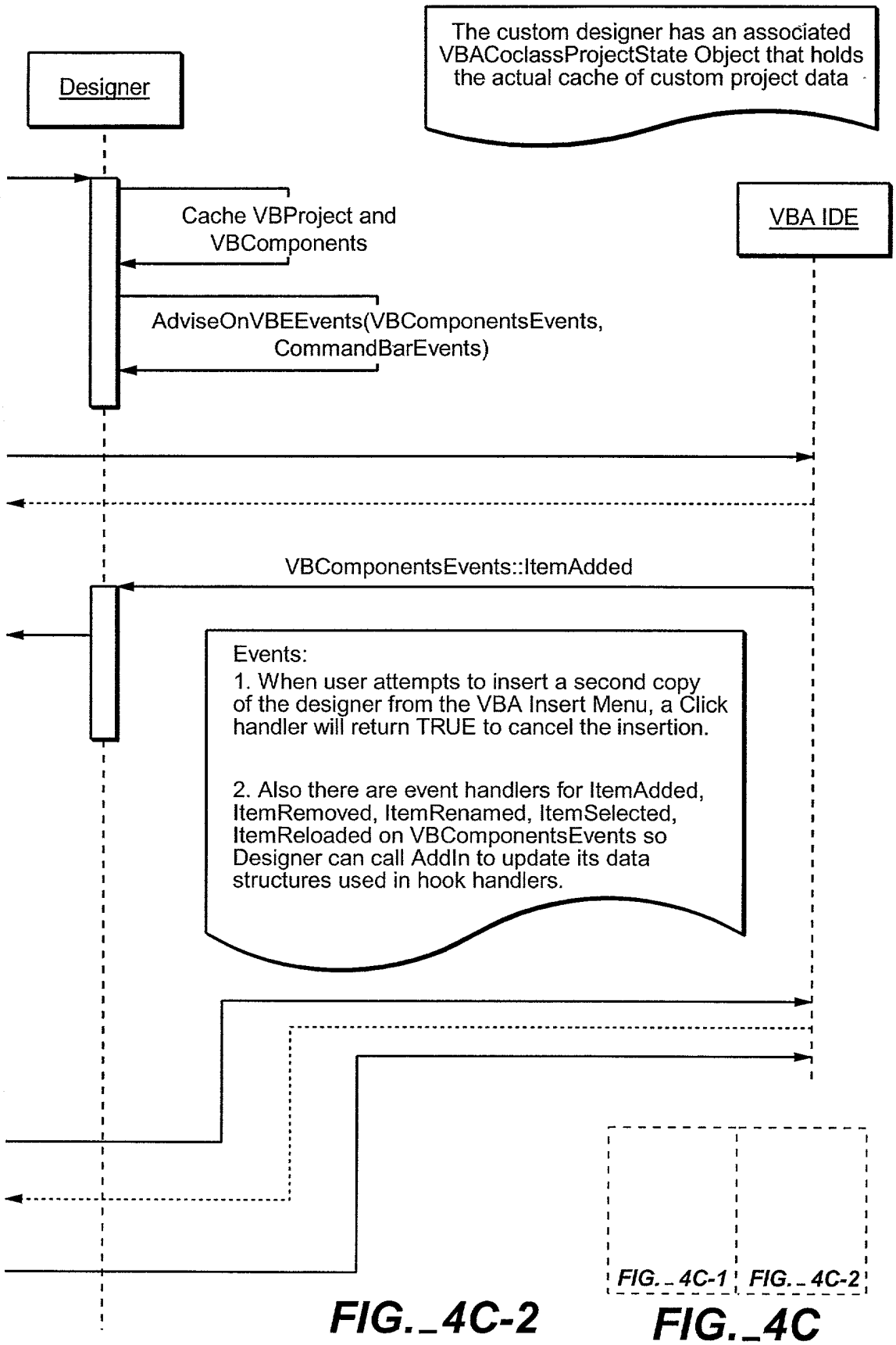


FIG._4C-2

FIG._4C-1 FIG._4C-2

FIG._4C

VBA Project Compilation & Save 11 / 27

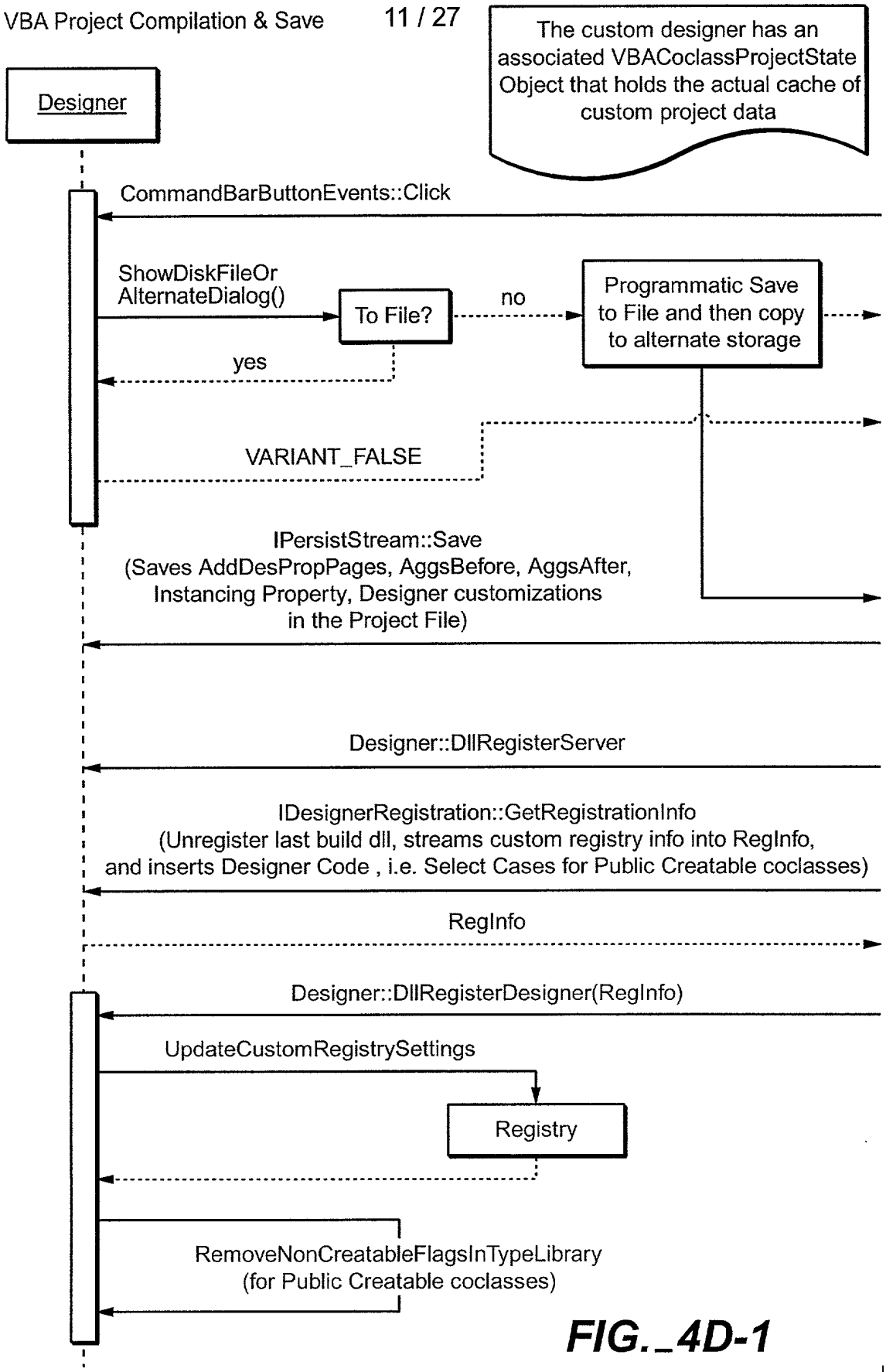


FIG._4D-1

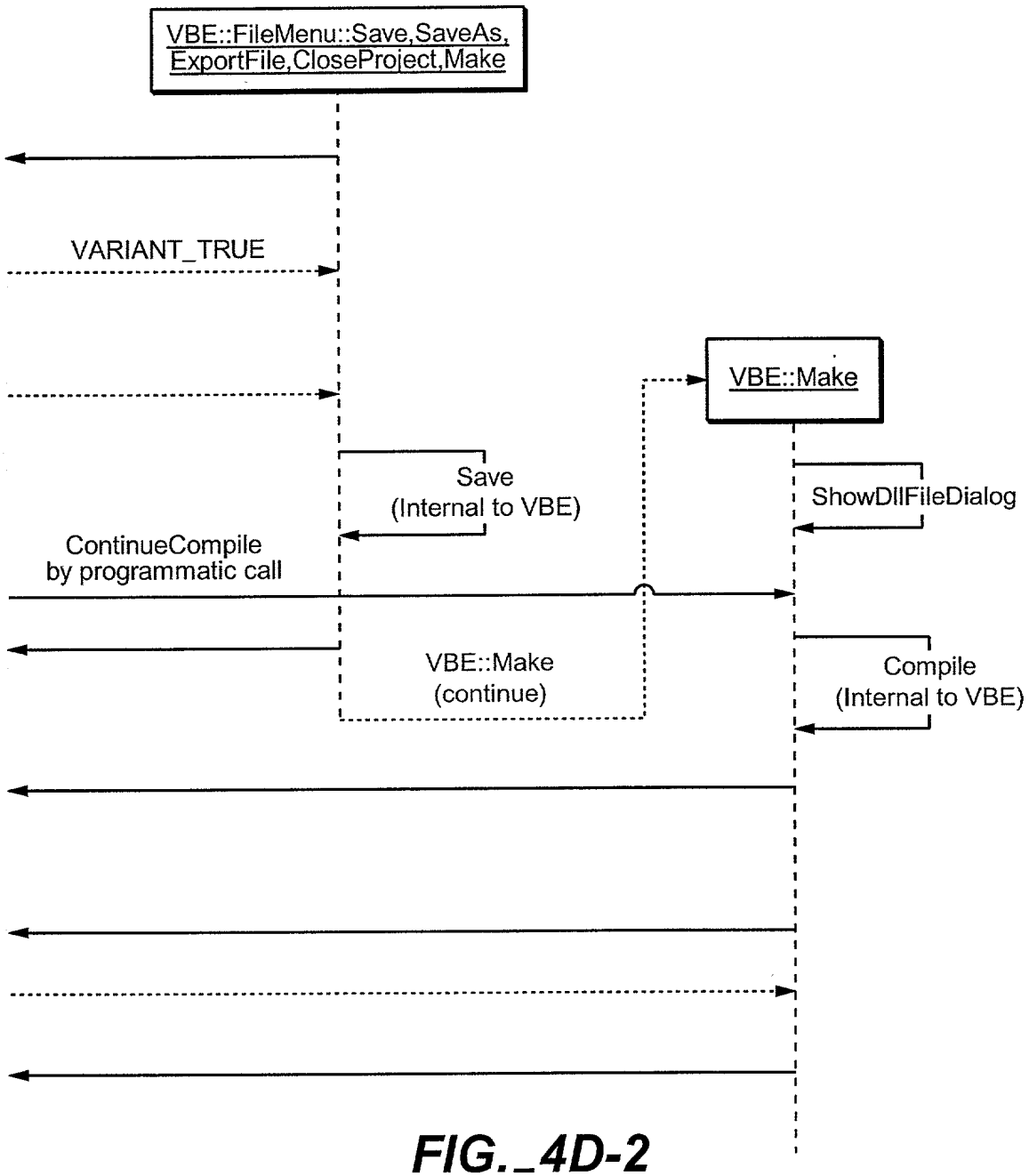


FIG._4D-2

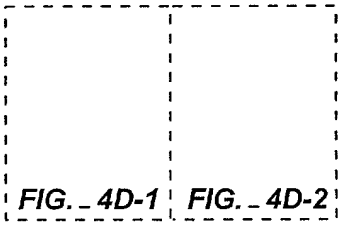
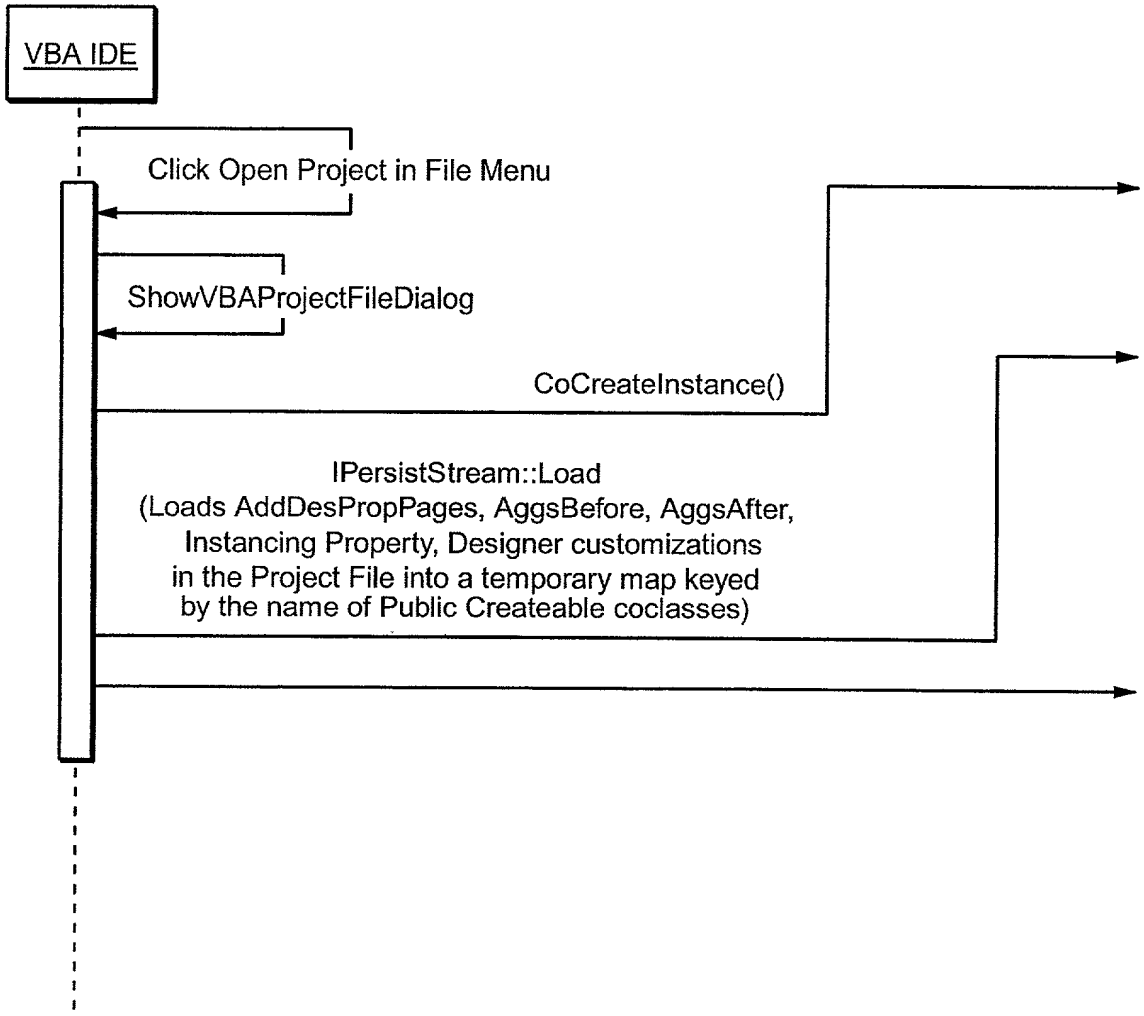


FIG._4D

VBA Project Open, Removal 13 / 27



When the user selects deleting a project from the VBA IDE, the VBProjectsEvents:: ProjItemRemoved handler in the Addin removes the project's custom settings in the data structures in the AddIn. Also when the user inserts/deletes a designer into/from a project, or inserts/deletes a class into/from a project, the VBCommandBarEvents:: Click event handler in the AddIn and designer, respectively, add to or delete from the data caches in the AddIn and designer.

FIG._4E-1

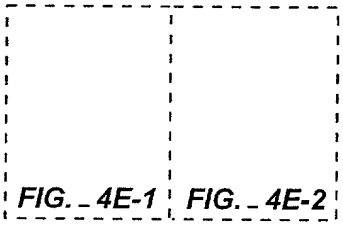


FIG._4E

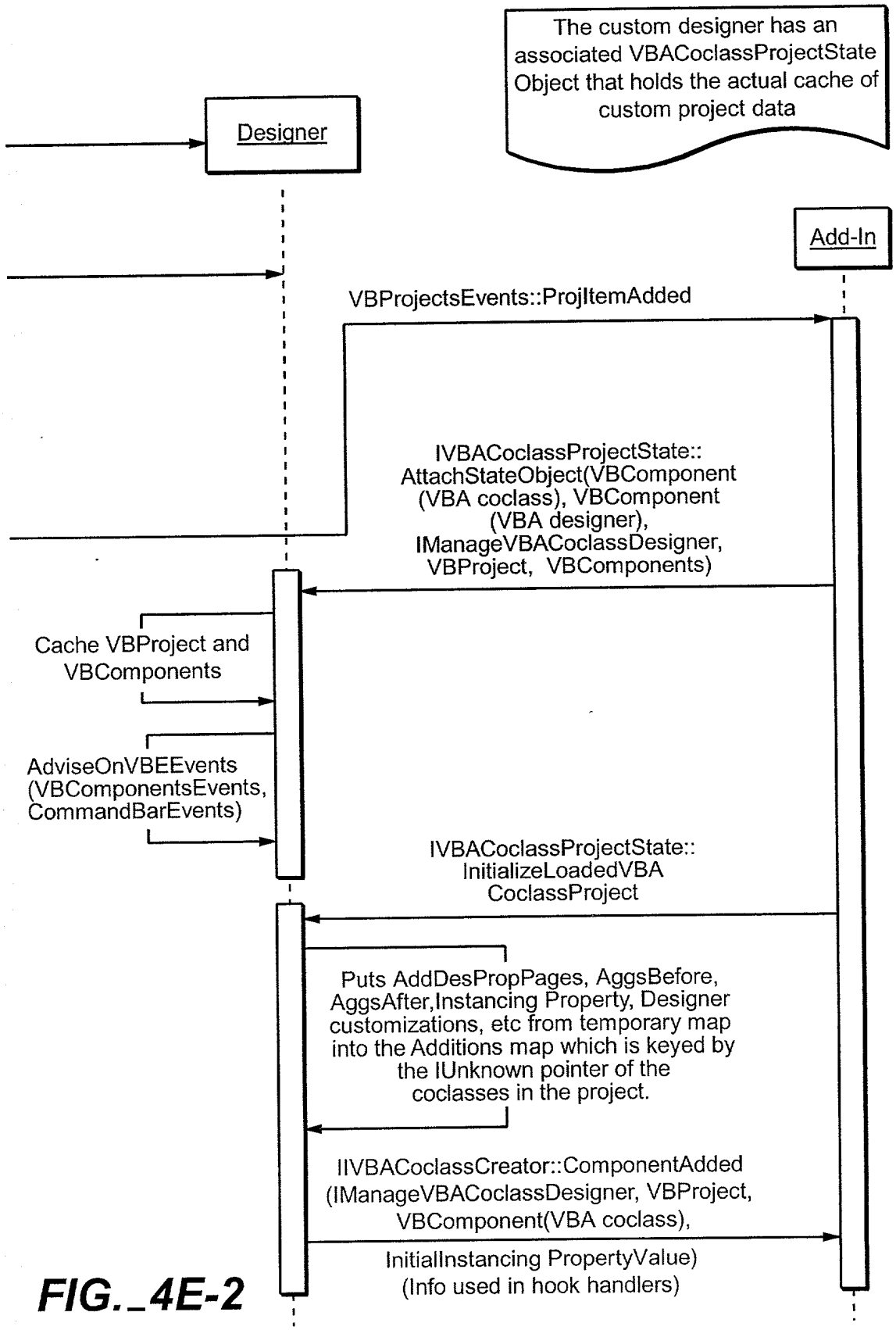

The custom designer has an associated VBACoclassProjectState Object that holds the actual cache of custom project data
Designer
Add-In
VBProjectsEvents::ProjItemAdded
IVBACoclassProjectState:: AttachStateObject(VBComponent (VBA coclass), VBComponent (VBA designer), IManageVBACoclassDesigner, VBProject, VBComponents)
Cache VBProject and VBComponents
AdviseOnVBEEvents (VBComponentsEvents, CommandBarEvents)
IVBACoclassProjectState:: InitializeLoadedVBA CoclassProject
Puts AddDesPropPages, AggsBefore, AggsAfter,Instancing Property, Designer customizations, etc from temporary map into the Additions map which is keyed by the IUnknown pointer of the coclasses in the project.
IIVBACoclassCreator::ComponentAdded (IManageVBACoclassDesigner, VBProject, VBComponent(VBA coclass), InitialInstancing PropertyValue) (Info used in hook handlers)

FIG._4E-2

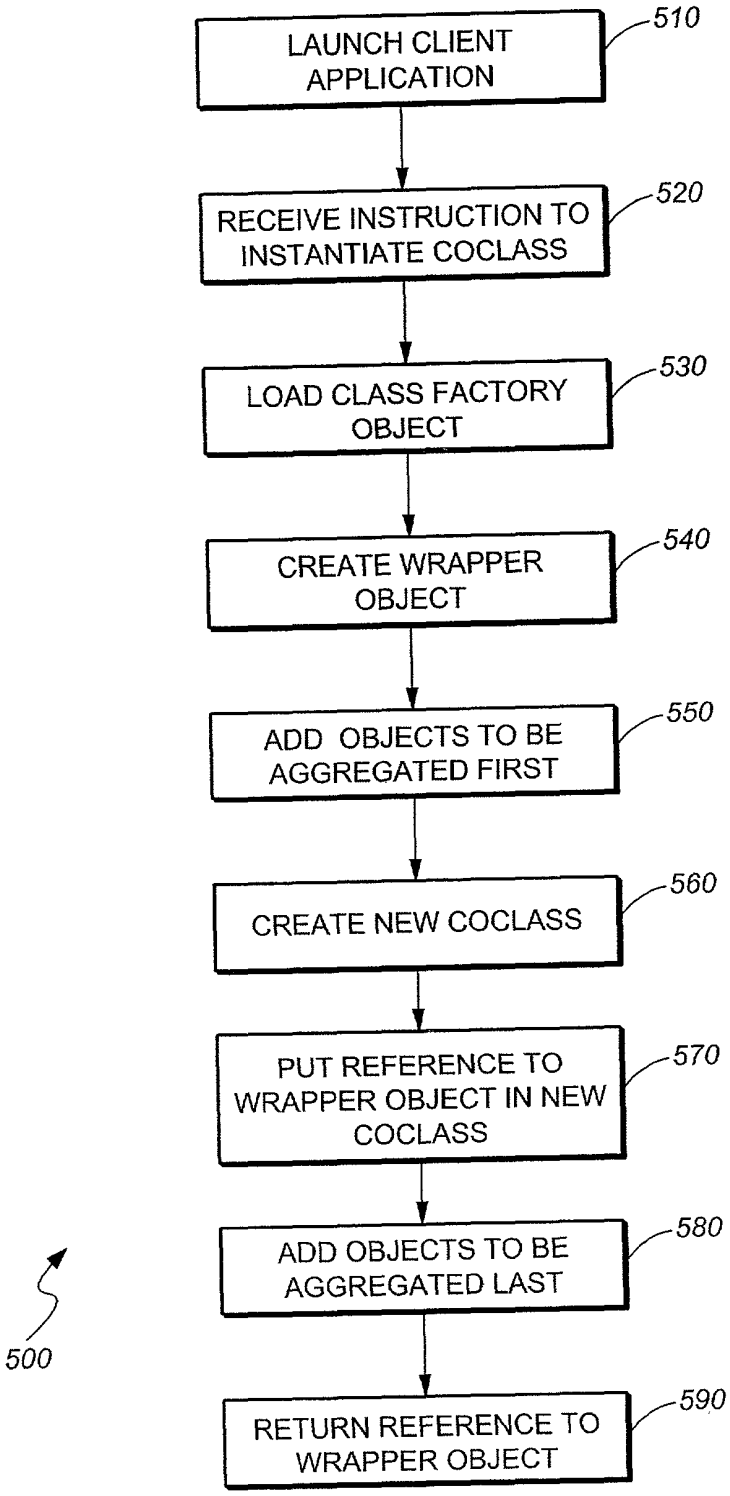

LAUNCH CLIENT APPLICATION
510
RECEIVE INSTRUCTION TO INSTANTIATE COCLASS
520
LOAD CLASS FACTORY OBJECT
530
CREATE WRAPPER OBJECT
540
ADD OBJECTS TO BE AGGREGATED FIRST
550
CREATE NEW COCLASS
560
PUT REFERENCE TO WRAPPER OBJECT IN NEW COCLASS
570
ADD OBJECTS TO BE AGGREGATED LAST
580
RETURN REFERENCE TO WRAPPER OBJECT
590
500

FIG._5

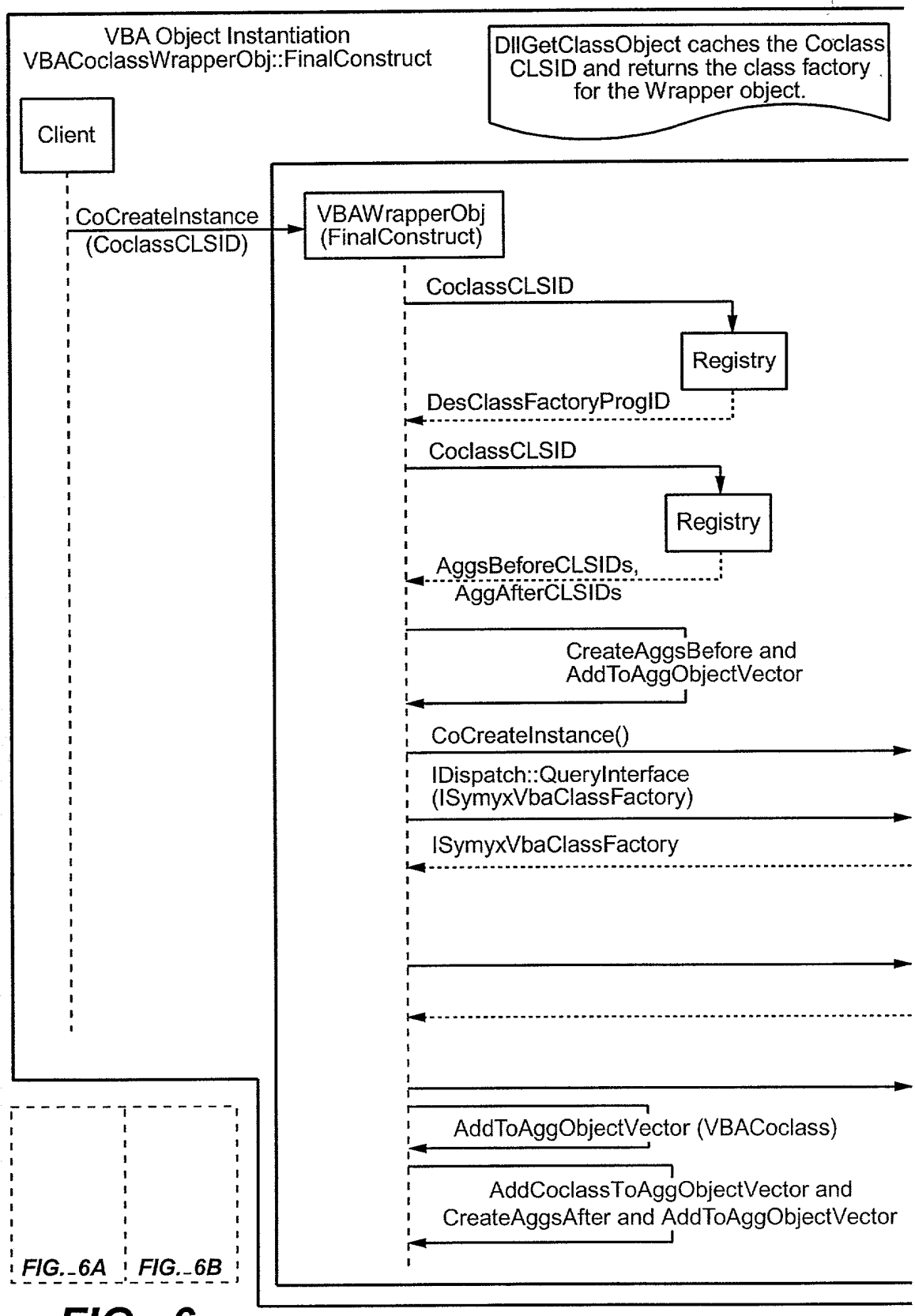


FIG._6A FIG._6B

FIG._6

FIG._6A

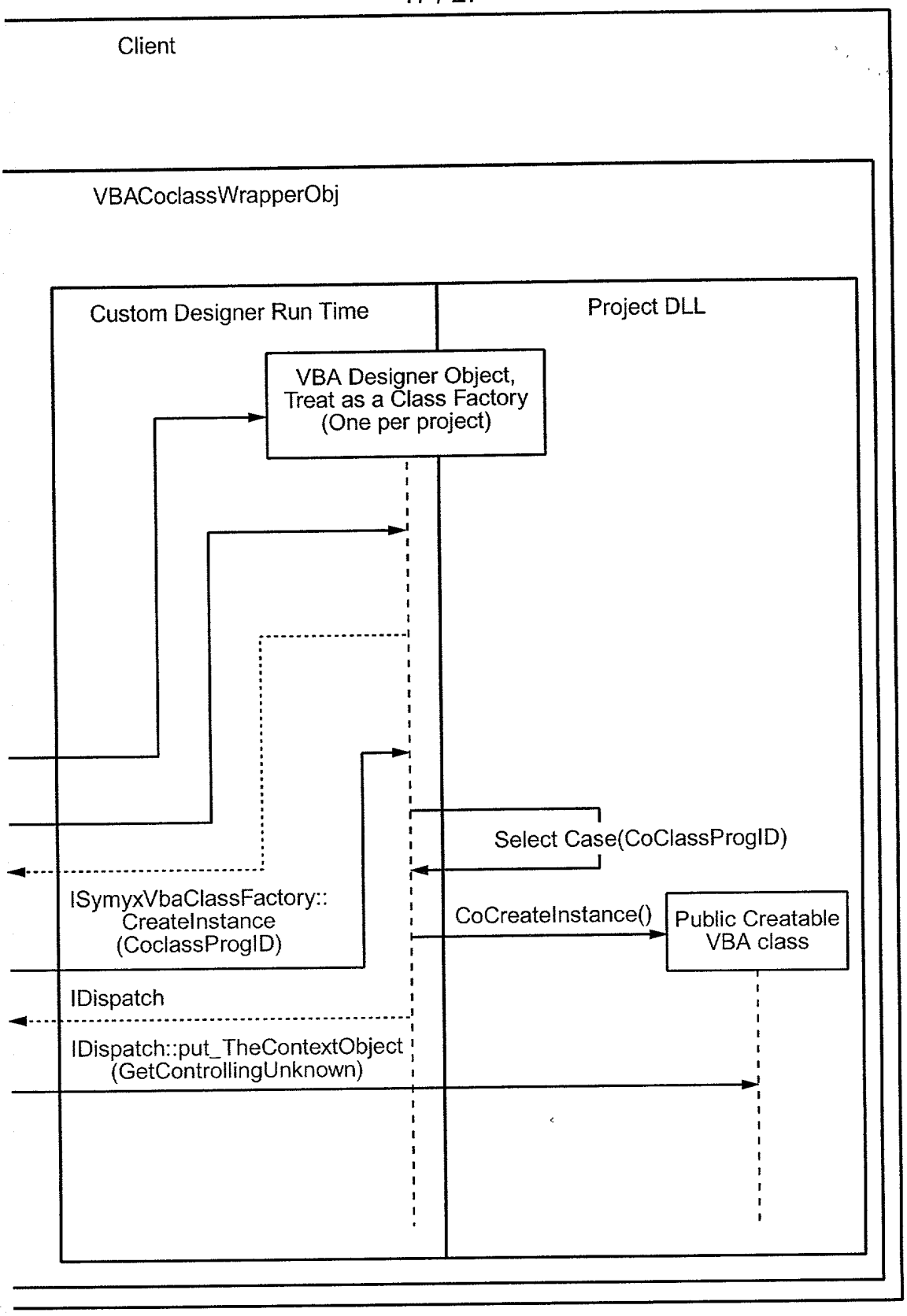

Client
VBACoclassWrapperObj
Custom Designer Run Time
Project DLL
VBA Designer Object,
Treat as a Class Factory
(One per project)
Select Case(CoClassProgID)
ISymyxVbaClassFactory::
CreateInstance
(CoclassProgID)
CoCreateInstance()
Public Creatable
VBA class
IDispatch
IDispatch::put_TheContextObject
(GetControllingUnknown)

FIG._6B

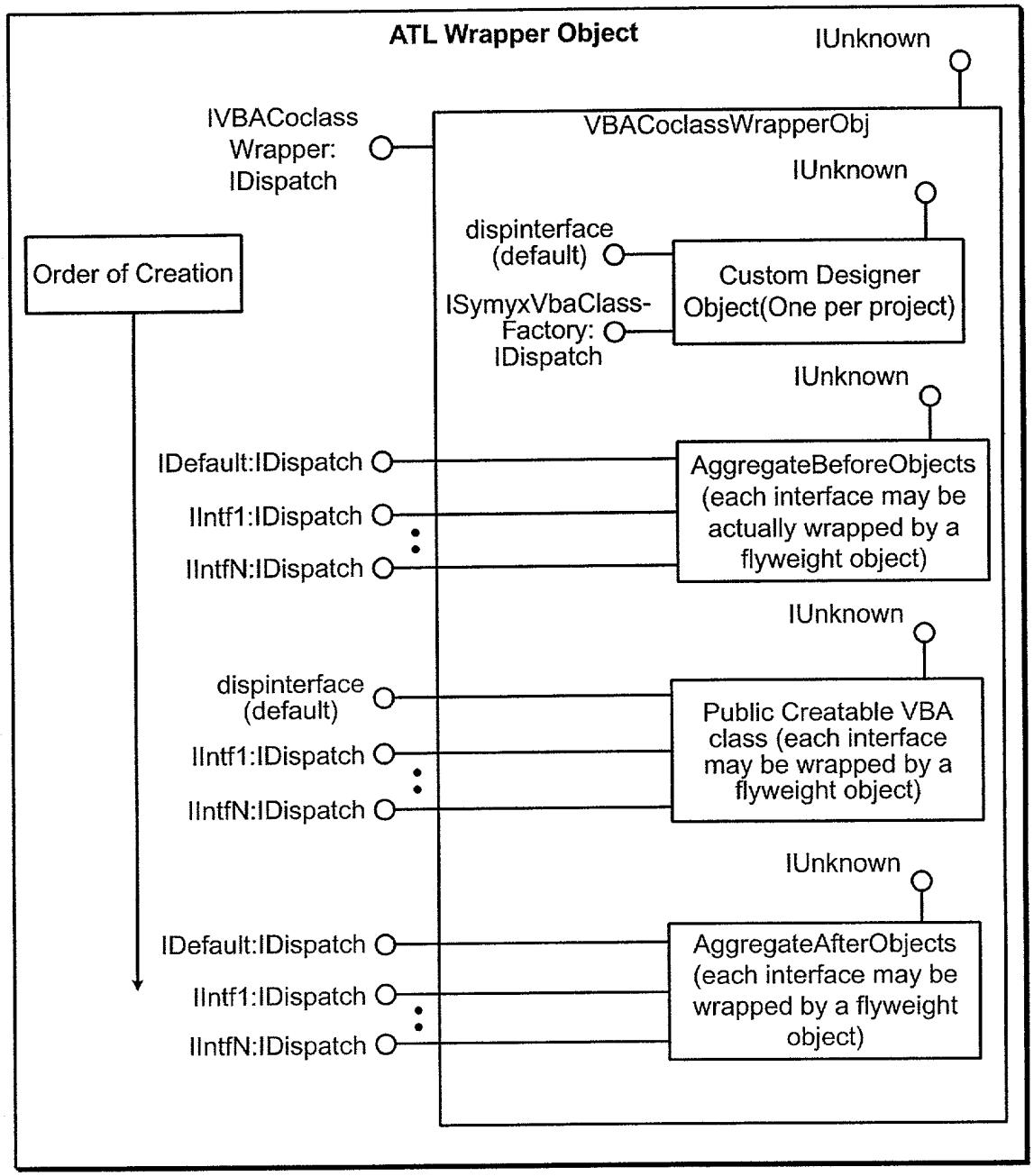

ATL Wrapper Object
IUnknown
IVBACoclass Wrapper: IDispatch
VBACoclassWrapperObj
IUnknown
dispinterface (default)
Custom Designer Object(One per project)
ISymyxVbaClass-Factory: IDispatch
Order of Creation
IUnknown
IDefault:IDispatch
IIntf1:IDispatch
IIntfN:IDispatch
AggregateBeforeObjects (each interface may be actually wrapped by a flyweight object)
IUnknown
dispinterface (default)
IIntf1:IDispatch
IIntfN:IDispatch
Public Creatable VBA class (each interface may be wrapped by a flyweight object)
IUnknown
IDefault:IDispatch
IIntf1:IDispatch
IIntfN:IDispatch
AggregateAfterObjects (each interface may be wrapped by a flyweight object)

FIG._7

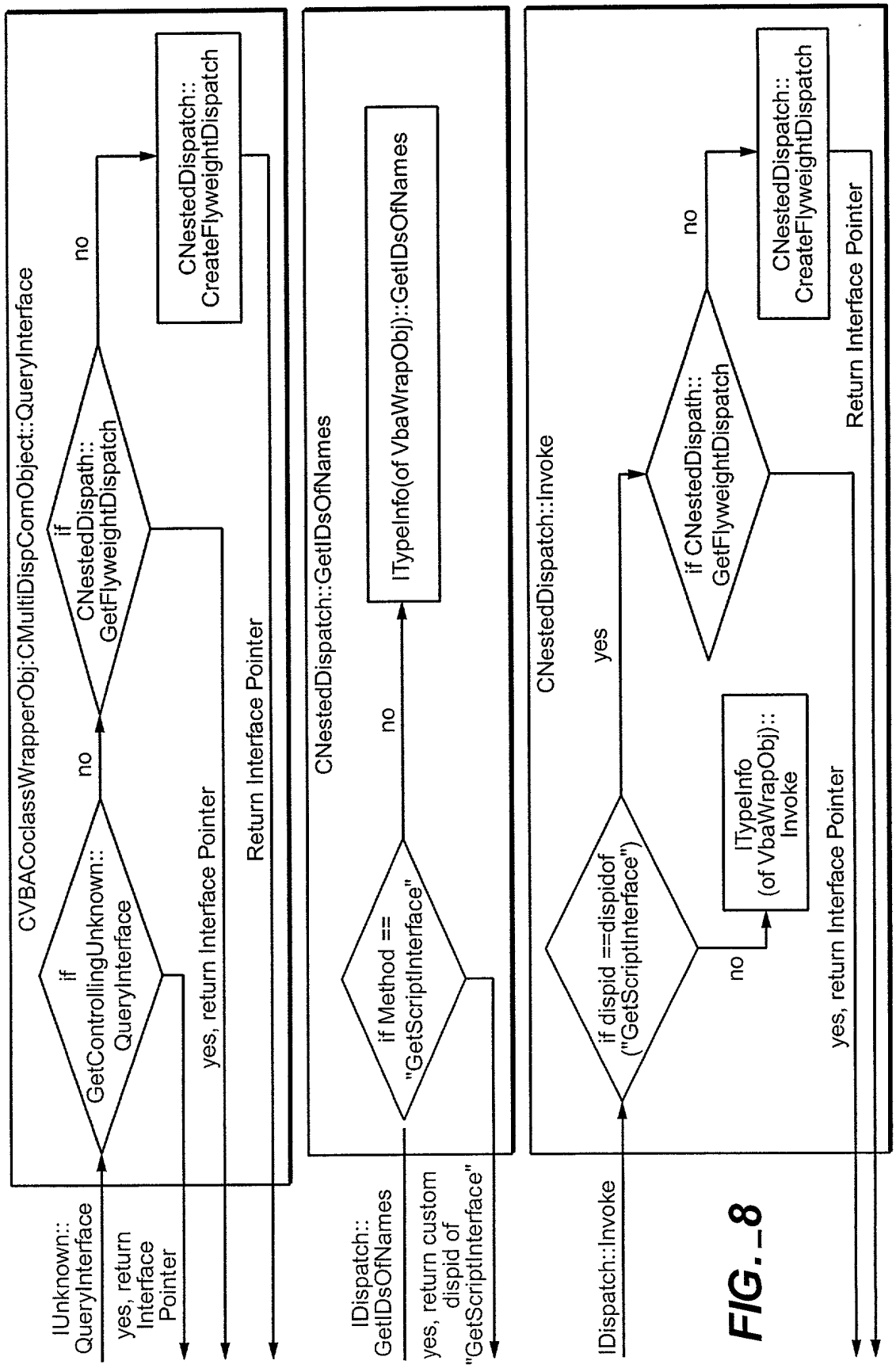

IUnknown::
QueryInterface
yes, return
Interface
Pointer
CVBACoclassWrapperObj:CMultiDispComObject::QueryInterface
if
GetControllingUnknown::
QueryInterface
no
if
CNestedDispath::
GetFlyweightDispatch
no
CNestedDispatch::
CreateFlyweightDispatch
yes, return Interface Pointer
Return Interface Pointer
IDispatch::
GetIDsOfNames
yes, return custom
dispid of
"GetScriptInterface"
CNestedDispatch::GetIDsOfNames
if Method ==
"GetScriptInterface"
no
ITypeInfo(of VbaWrapObj)::GetIDsOfNames
IDispatch::Invoke
CNestedDispatch::Invoke
if dispid ==dispidof
("GetScriptInterface")
yes
no
ITypeInfo
(of VbaWrapObj)::
Invoke
if CNestedDispath::
GetFlyweightDispatch
no
CNestedDispatch::
CreateFlyweightDispatch
Return Interface Pointer
yes, return Interface Pointer

FIG._8

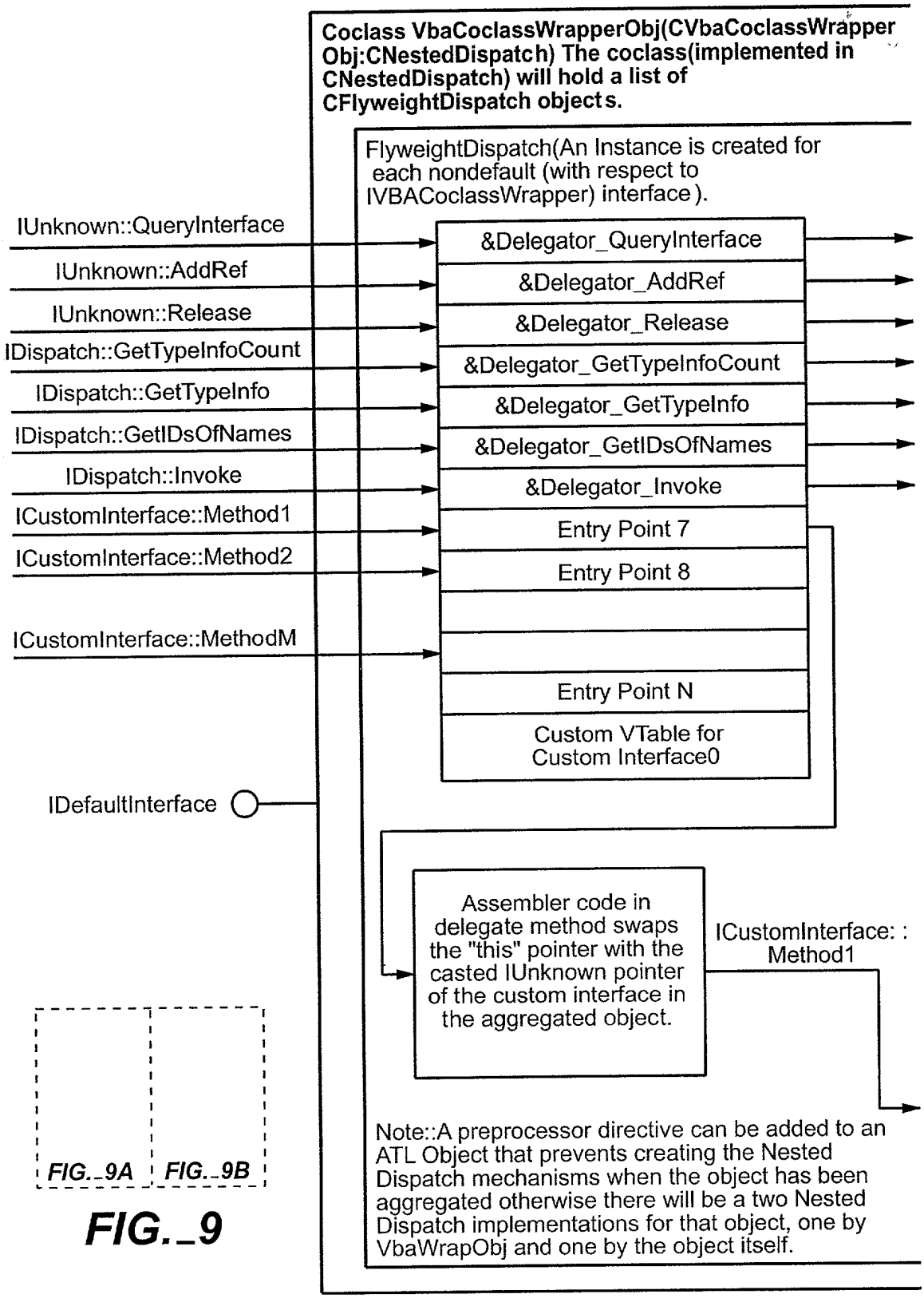


FIG._9A FIG._9B

FIG._9

FIG._9A

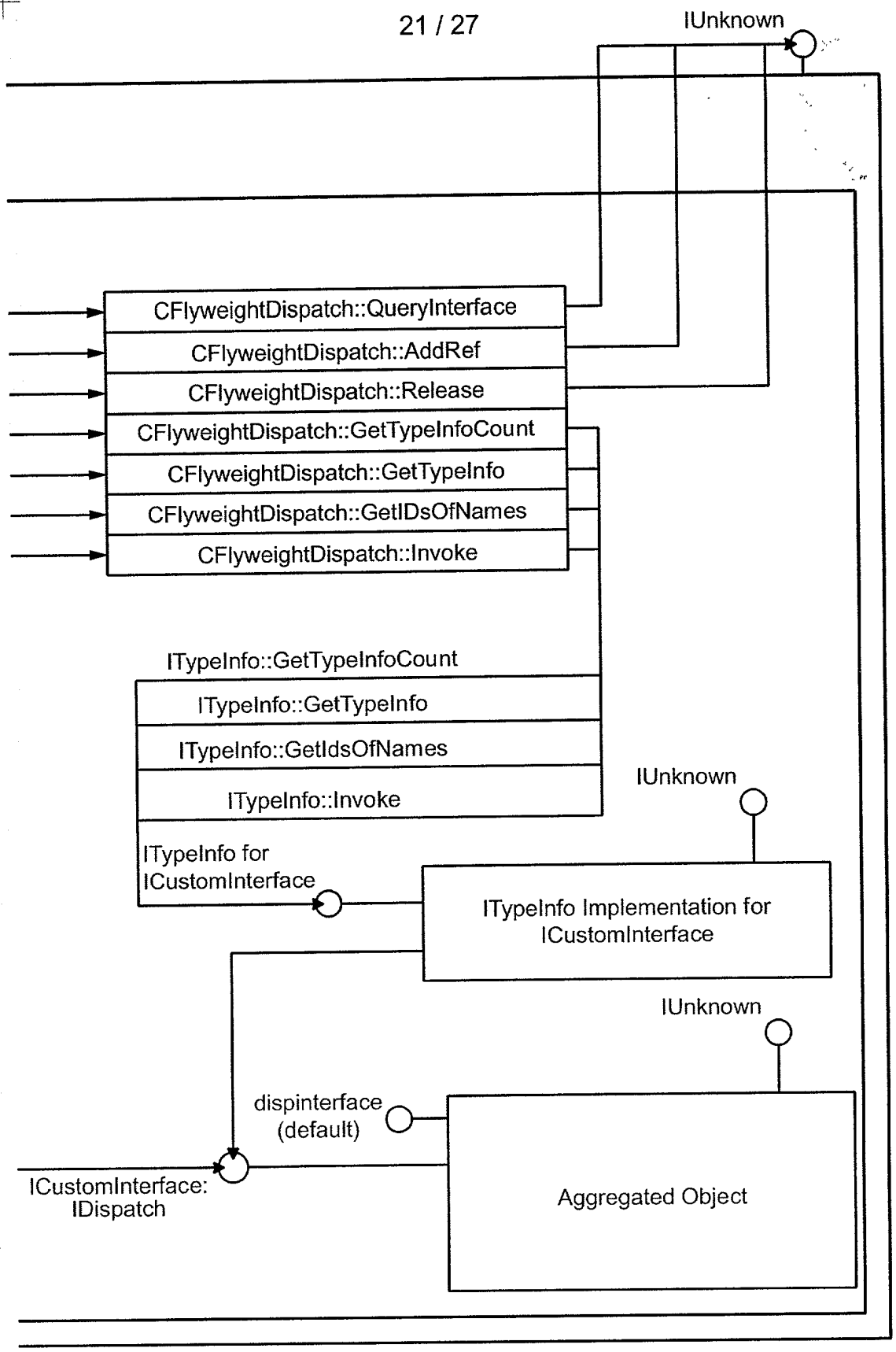

IUnknown
CFlyweightDispatch::QueryInterface
CFlyweightDispatch::AddRef
CFlyweightDispatch::Release
CFlyweightDispatch::GetTypeInfoCount
CFlyweightDispatch::GetTypeInfo
CFlyweightDispatch::GetIDsOfNames
CFlyweightDispatch::Invoke
ITypeInfo::GetTypeInfoCount
ITypeInfo::GetTypeInfo
ITypeInfo::GetIdsOfNames
ITypeInfo::Invoke
IUnknown
ITypeInfo for
ICustomInterface
ITypeInfo Implementation for
ICustomInterface
IUnknown
dispinterface
(default)
ICustomInterface:
IDispatch
Aggregated Object

FIG._9B

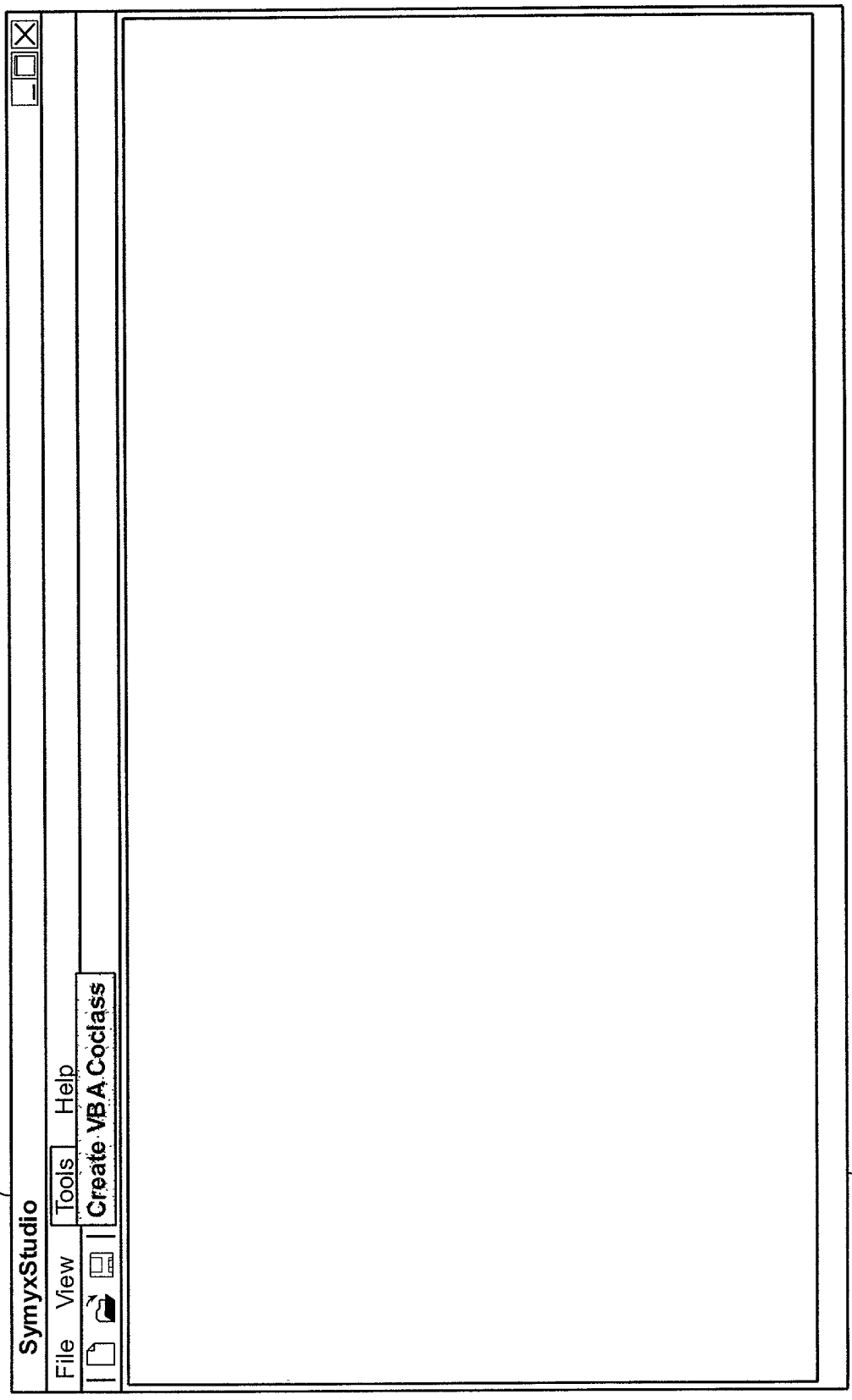

SymyxStudio
File View Tools Help
Create VB A Coclass
1010
1000

FIG._10A

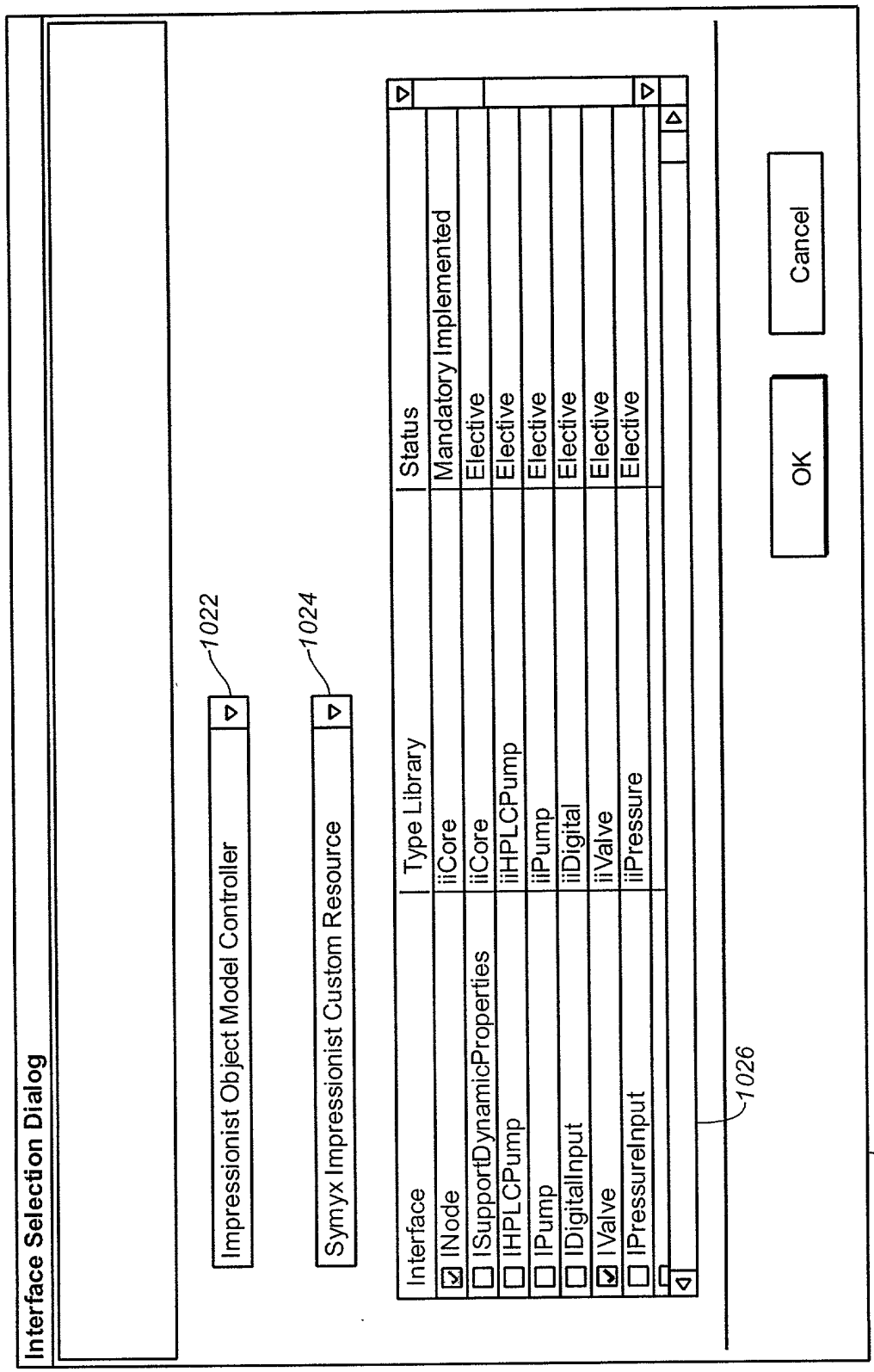

Interface Selection Dialog
Impressionist Object Model Controller
1022
Symyx Impressionist Custom Resource
1024
Interface
Type Library
Status
INode
iiCore
Mandatory Implemented
ISupportDynamicProperties
iiCore
Elective
IHPLCPump
iiHPLCPump
Elective
IPump
iiPump
Elective
IDigitalInput
iiDigital
Elective
IValve
iiValve
Elective
IPressureInput
iiPressure
Elective
1026
OK
Cancel
1020

FIG._10B

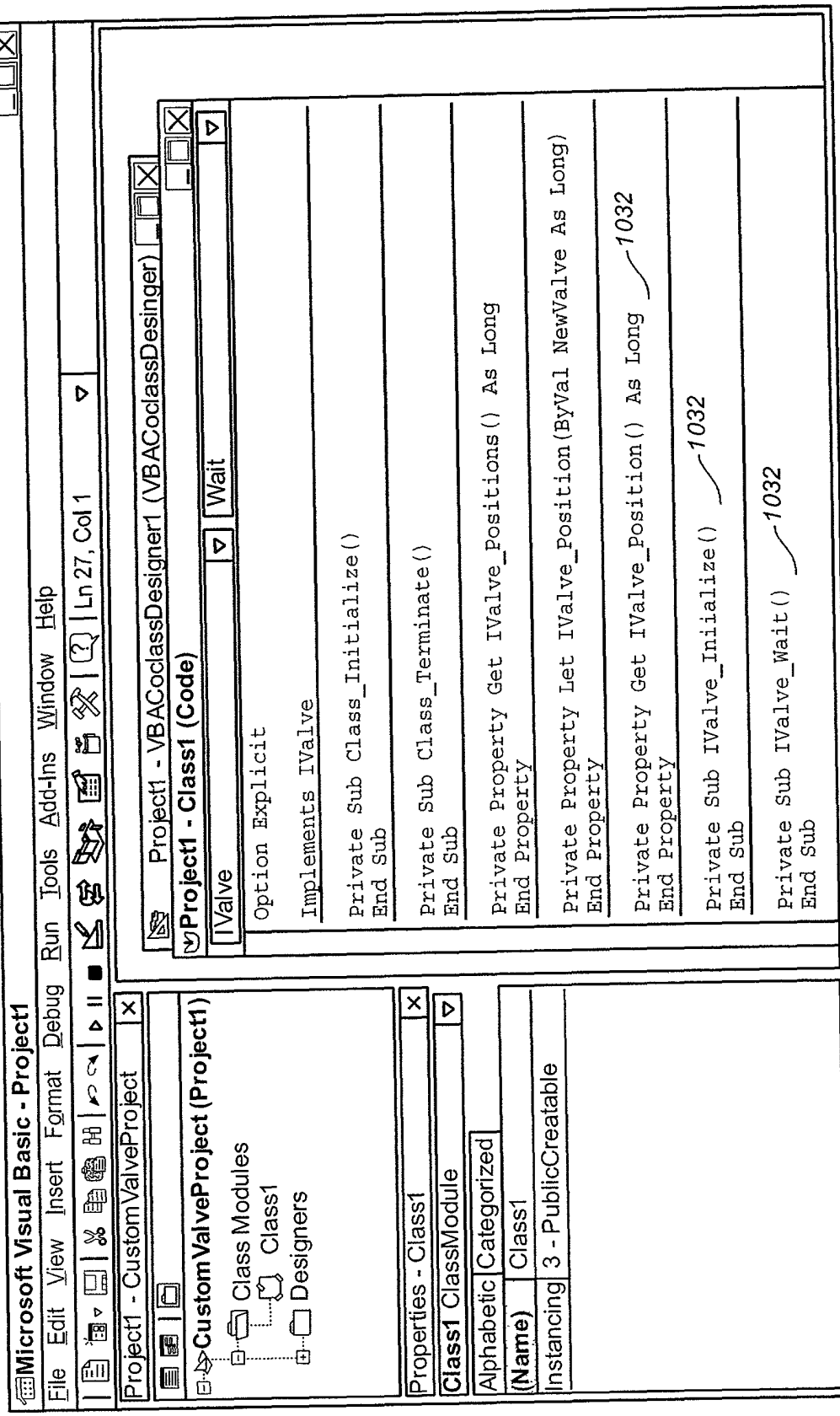

Microsoft Visual Basic - Project1
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 27, Col 1
Project1 - CustomValveProject
CustomValveProject (Project1)
Class Modules
Class1
Designers
Properties - Class1
Class1 ClassModule
Alphabetic Categorized
(Name) Class1
Instancing 3 - PublicCreatable
Project1 - VBACoclassDesigner1 (VBACoclassDesinger)
Project1 - Class1 (Code)
IValve
Wait
Option Explicit
Implements IValve
Private Sub Class_Initialize()
End Sub
Private Sub Class_Terminate()
End Sub
Private Property Get IValve_Positions() As Long
End Property
Private Property Let IValve_Position(ByVal NewValve As Long)
End Property
Private Property Get IValve_Position() As Long
End Property
Private Sub IValve_Initialize()
End Sub
Private Sub IValve_Wait()
End Sub
1032
1032
1032
1030

FIG._10C

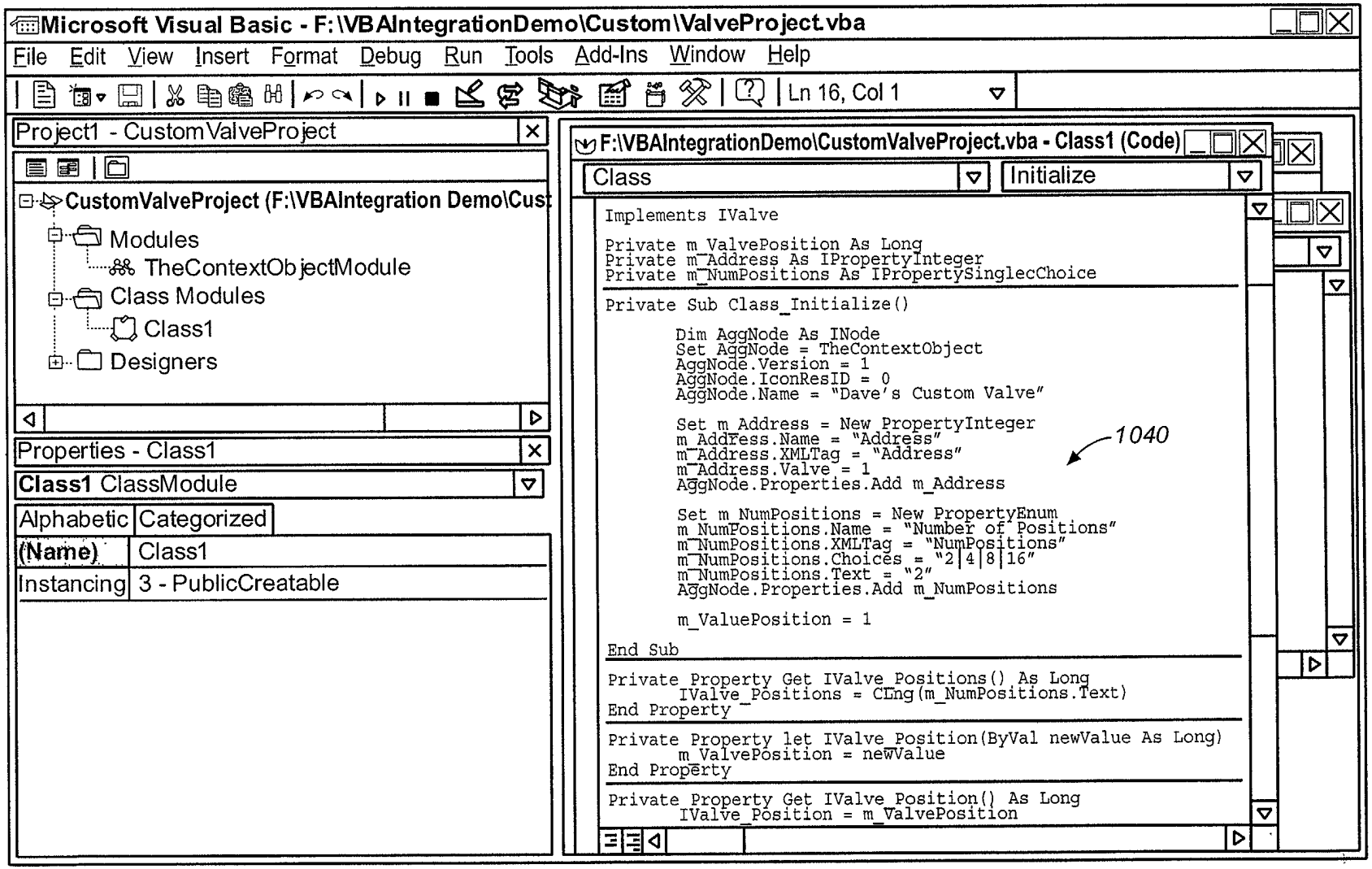

Microsoft Visual Basic - F:\VBAIntegrationDemo\Custom\ValveProject.vba
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 16, Col 1
Project1 - CustomValveProject
CustomValveProject (F:\VBAIntegration Demo\Cust
Modules
TheContextObjectModule
Class Modules
Class1
Designers
Properties - Class1
Class1 ClassModule
Alphabetic
Categorized
(Name)
Class1
Instancing
3 - PublicCreatable
F:\VBAIntegrationDemo\CustomValveProject.vba - Class1 (Code)
Class
Initialize
Implements IValve
Private m_ValvePosition As Long
Private m_Address As IPropertyInteger
Private m_NumPositions As IPropertySinglecChoice
Private Sub Class_Initialize()
Dim AggNode As INode
Set AggNode = TheContextObject
AggNode.Version = 1
AggNode.IconResID = 0
AggNode.Name = "Dave's Custom Valve"
Set m_Address = New PropertyInteger
m_Address.Name = "Address"
m_Address.XMLTag = "Address"
m_Address.Valve = 1
AggNode.Properties.Add m_Address
1040
Set m_NumPositions = New PropertyEnum
m_NumPositions.Name = "Number of Positions"
m_NumPositions.XMLTag = "NumPositions"
m_NumPositions.Choices = "2|4|8|16"
m_NumPositions.Text = "2"
AggNode.Properties.Add m_NumPositions
m_ValuePosition = 1
End Sub
Private Property Get IValve_Positions() As Long
IValve_Positions = CLng(m_NumPositions.Text)
End Property
Private Property let IValve_Position(ByVal newValue As Long)
m_ValvePosition = newValue
End Property
Private Property Get IValve_Position() As Long
IValve_Position = m_ValvePosition

FIG._10D

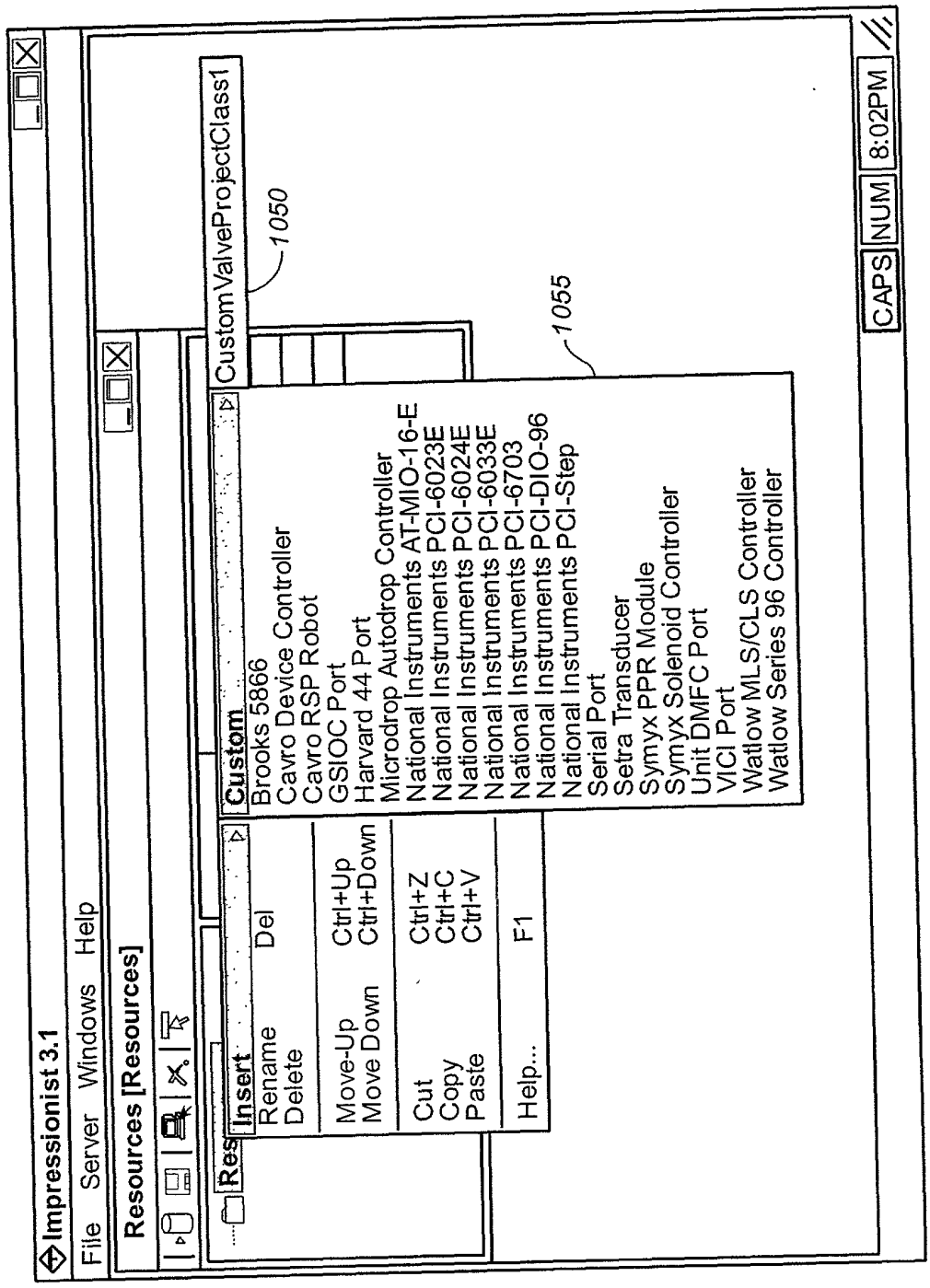

Impressionist 3.1
File Server Windows Help
Resources [Resources]
Res
Insert
Rename Del
Delete
Move-Up Ctrl+Up
Move Down Ctrl+Down
Cut Ctrl+Z
Copy Ctrl+C
Paste Ctrl+V
Help... F1
Custom
Brooks 5866
Cavro Device Controller
Cavro RSP Robot
GSIOC Port
Harvard 44 Port
Microdrop Autodrop Controller
National Instruments AT-MIO-16-E
National Instruments PCI-6023E
National Instruments PCI-6024E
National Instruments PCI-6033E
National Instruments PCI-6703
National Instruments PCI-DIO-96
National Instruments PCI-Step
Serial Port
Setra Transducer
Symyx PPR Module
Symyx Solenoid Controller
Unit DMFC Port
VICI Port
Watlow MLS/CLS Controller
Watlow Series 96 Controller
CustomValveProjectClass1
1050
1055
CAPS NUM 8:02PM

FIG._10E

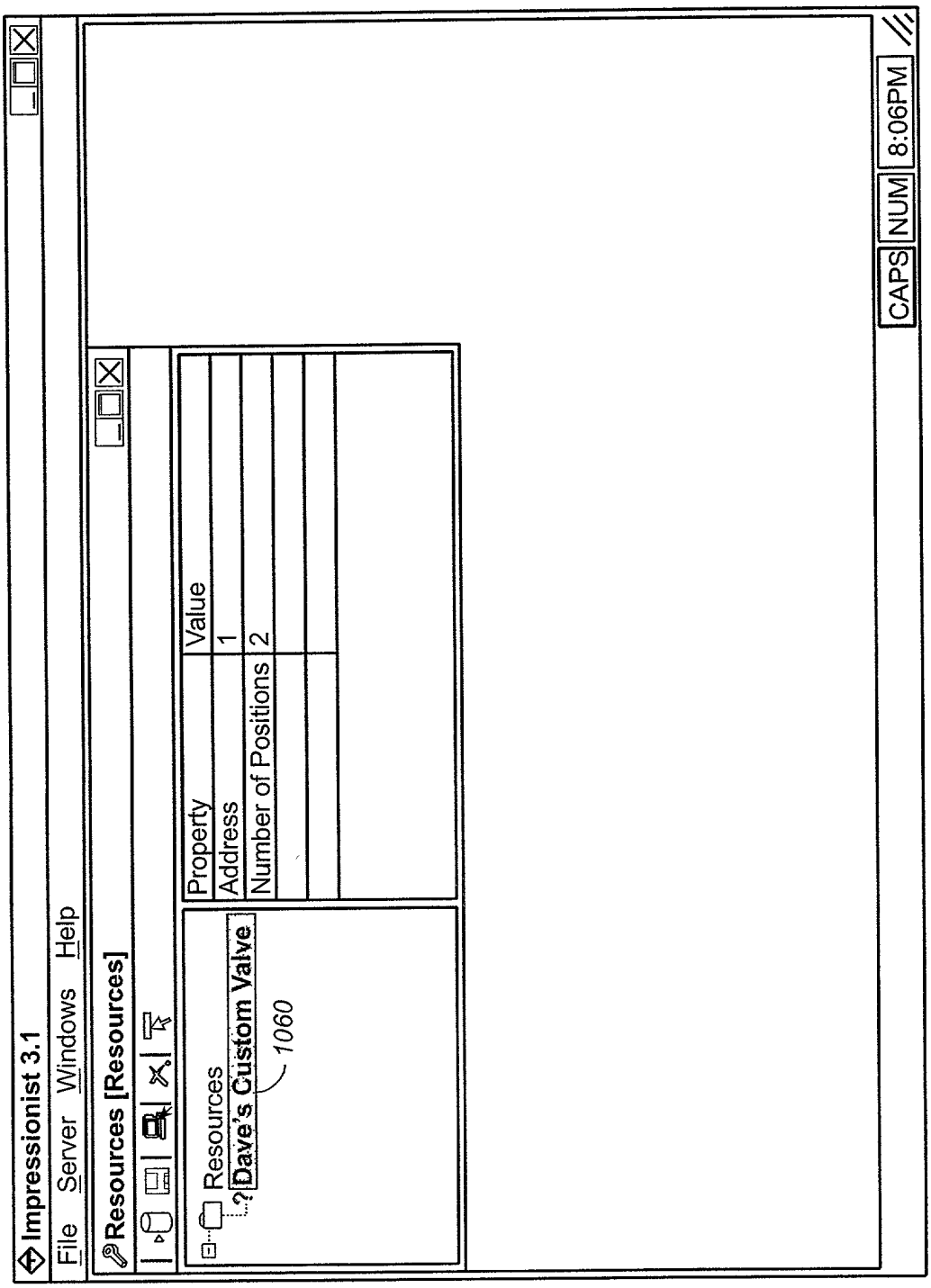

Impressionist 3.1
File Server Windows Help
Resources [Resources]
Resources
Dave's Custom Valve
1060
Property
Value
Address
1
Number of Positions
2
CAPS
NUM
8:06PM

FIG._10F